FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Emotional Agents for Shooter Games: Understanding How Players' Emotional Profiles Influence Game Playouts

Bruno Xavier Faria Tavares

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Eugénio da Costa Oliveira

Second Supervisor: Pedro Gonçalo Ferreira Alves Nogueira

July 07, 2015

Emotional Agents for Shooter Games: Understanding How Players' Emotional Profiles Influence Game Playouts

Bruno Xavier Faria Tavares

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Doctor António Fernando Vasconcelos Cunha Castro Coelho
External Examiner: Doctor Pedro Moreira
Supervisor: Doctor Eugénio da Costa Oliveira
July 07, 2015

Abstract

Gameplay testing still suffers from inefficient feedback assimilation, not only due to the subjective nature of the retrieved information but also due to the amount of time required to retrieve it. This work attempts to ameliorate that issue by automating the testing process without losing all emotional data. We aim to achieve this by modelling an agent to replicate expected human behaviours. The models are created based on previously collected data from actual gameplay sessions, translated into Arousal and Valence values, through fuzzy clustering. With this proof of concept, we expect to emulate human behaviours in a satisfactory manner and evaluate the usefulness of this method as a quality assurance tool.

Acknowledgements

I would like to start by expressing my gratitude to my supervisors, Prof. Eugénio da Costa Oliveira and Msc. Pedro Gonçalo Ferreira Alves Nogueira, for believing in me despite adverse situations. Were it not for them, this project would not be here today.

A special thank you to the GameDev subreddit, which proved an invaluable resource throughout the development stages.

I also have to thank everyone who took time of their schedule to participate on my survey.

Finally, to all my friends who supported me thus far, I hope we all manage to reach for the stars.

Bruno Xavier Faria Tavares

"The turning of the tide always begins with one soldier's decision to head back into the fray."

Glory Seeker, Wizards of the Coast

Contents

CONTENTS

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

Abbreviations

AI	Artificial Intelligence
A/V	Arousal/Valence
BDI	Belief-Desire-Intention
BT	Behaviour Tree
FPS	First Person Shooter
GDM	Gaussian Distance Matrix
FSM	Finite State Machines
HFSM	Hierarchical State Machines
NPC	Non-Player Character

Chapter 1

Introduction

Artificial Intelligence has recently become a popular research topic, especially video games[AS10], when compared with how long it has been used, some applications of it more complex than others. When it comes to computer science, Artificial Intelligence has mostly been focused on applying different algorithms to solve various problems that would be difficult for a human to solve in the necessary time frame, either assisting human decision or replacing it entirely. Systems implemented based on this range from "expert systems", which possess knowledge over a specific domain to assist human decisions with their own decisions based mainly on if-else rules, to robot platforms, which can have a variety of routines that can be configured to be executed on various conditions. Examples of these systems are clinical decision support systems for medical diagnosis, image recognition on computer vision systems, natural language processing for customer service systems, and many others.

1.1 Context

In computer science, one of the fields that is closely connected to AI developments is video game development. Artificial Intelligence on this kind of software has been present since the development of the first video games, notably Pong, which supported a single player experience where the computer would control the opposing paddle. Early AI agents in video games had to be very simplistic, due to the low computing power of the time and the still present need for results in real time, consisting mostly of very simple conditions. As computing power increased, especially during the last few years, video game developers have harnessed the power of AI to create smarter opponents, as well as more realistic simulations of a given world, thanks to the agent architectures that have become popular and are well suited for video games. These can be seen in greater detail on games such as The Sims (Maxis,2000), which attributes a certain personality to every agent which incentive certain actions in opposition to others while giving them desires and needs, as

well as the Nemesis System present in Shadow of Mordor (Monolith Productions, 2014), that despite being a more open ended experience, presents many enemy agents who react to the player's actions differently and evolve through the enemy's hierarchy differently, possibly fighting among themselves in order to become stronger. One very particular example is F.E.A.R. (Moonlight Productions,2005), that was largely praised over his ruthless AI that used the opponent's agents in a tactical way to try and hamper the progress of the player, taking advantage of the players' lack of knowledge of the map to flank him and do coordinated strikes. This behaviour is considered emergent gameplay, since the AI agents were not developed to be that complex[Ork06]: the interaction between the simple rules used at the time made the system appear much smarter than it actually was.

Shooter games tend to be action-oriented experiences, sometimes sacrificing other elements, such as story and character development, to improve on the mechanics and provide a more cinematic experience. This type of games normally uses AI's for opponents that attempt to eliminate the player before he can reach the goal, by having them patrol certain areas and using cover effectively, as well as allies which assist the player as he moves through the level. Depending on the opponent type, we have a multitude of possible answers to the players' actions, such as shooting, charging at the player, taking cover and setting up a defensive position, among others.

Survival/Horror games, however, tend to focus more on the players' emotions. By creating stressful and unconformable situations, they urge the player forward in the hopes of reaching a better location. While the opponent AI is still relevant, it is not as important as the context around it: using a number of techniques, such as environment manipulation and resource scarcity, even a simple opponent might trigger a very strong affective response from the player.

1.2 Motivation

This thesis attempts to create an AI agent that can mimic human reactions to a predetermined set of events, allowing it to be used as an assisting tool during game development. The advantages of such are twofold: on one hand, it allows developers to test iteratively without requiring a full QA cycle, which enables small adjustments to occur more frequently and with less costs than previously. On the other hand, it increases the amount of test data that reaches the game developer, which facilitates adjustments to different game aspects by taking into consideration the possible emotional states of the player. As such, these advantages will allow for a better game experience for the player, and a shorter development time for the game itself, while also making the gameplay experience closer to what the designer intended.

1.3 Goals

The main objective of this thesis is to create an AI agent which is able to do the following:

- Navigate through a given world, based on the information he has.
- React to a set of game events emitted by the game world.
- Alter his behaviour according to his emotional profile and the events he is subject to, causing sub-optimal behaviour in a game sense.
- be quickly altered to force or inhibit certain behaviours during the simulation while it is running.

To accomplish this goal, we intend to build a high-level simulator for a shooter game which has a number of predetermined events in order to test our agent model. This simulator will be based on the same game as the data the model is built upon to simplify event reactions, and will retrieve statistical data (such as accuracy, time taken to completion, favoured actions) to serve as comparison between human players and agents. As such, the simulator should be playable by human subjects as well. This also allows us to record playthroughs for a more subjective analysis, based on human opinions on which recordings correspond to a human subject.

1.4 Structure of the Thesis

This thesis contains 4 chapters beyond the introduction. In chapter 2, we will take a closer look at AI, specifically how to model human behaviour in such a way that can be represented through an AI agent, in addition to exploring different agent architectures. In chapter 3, we will present how we implemented the simulator, in addition to the player agent. We'll also discuss the models created for the agents, as well as the data they are based upon. In chapter 4, we will discuss the tests performed and their results. Finally, in chapter 5, we will evaluate whether or not our goals have been achieved and explore possibilities for future improvements.

Chapter 2

State of the Art

We will now begin the literary review, referencing relevant works to what our goals are. We'll start by looking at the efforts to model human behaviour made throughout the years. We will then look at how AI agents and how they can be implemented, focusing on emotional agents.

2.1 Model Behaviour

Over the years there have been numerous variations on how to model human behaviour. This desire to model behaviour began with the fields of Psychology, Neurology, Philosophy, and Cognitive Sciences to model the mind, and emotions are an important aspect of human behaviour. However, since the psychology of emotions is still evolving, creating a computational model that can accurately portray human emotions becomes difficult.[MZ07] This doesn't stop attempts to achieve it. One problem that stands out is the lack of consensus on the definition of emotion[GM04], or the range of phenomena within the domain of emotion research, that each model proposal has its own theory it accepts and builds upon.

Before examining models, it's important to understand where the data for these models is gathered. While there are many alternatives, such as questionnaires and gameplay analysis, these are always intrusive and subject to some bias, so one of the most recent practices is to recover physiological data from the subjects during gameplay sessions. A number of these can be seen in figure 2.1. This has multiple advantages, since it's possible to retrieve unbiased information as well as the important ability to recover continuous data, which enables a continuous model of the user's emotional state. This is especially relevant on play systems, because the quality of the experience of play is far more important than the mechanical performance during play[Man08]. We used this process to gather the data used for the creation of our models.

After data recovery, each model has a specific variation on how to handle the data. An example on how to accomplish this is converting the data into a set of dynamic models, such as Kalman filters, and sequence them with a Markov chain. This chain can then be used to recognize behaviours from sensory data, similar to the data used to create the model in the first place, and predict future

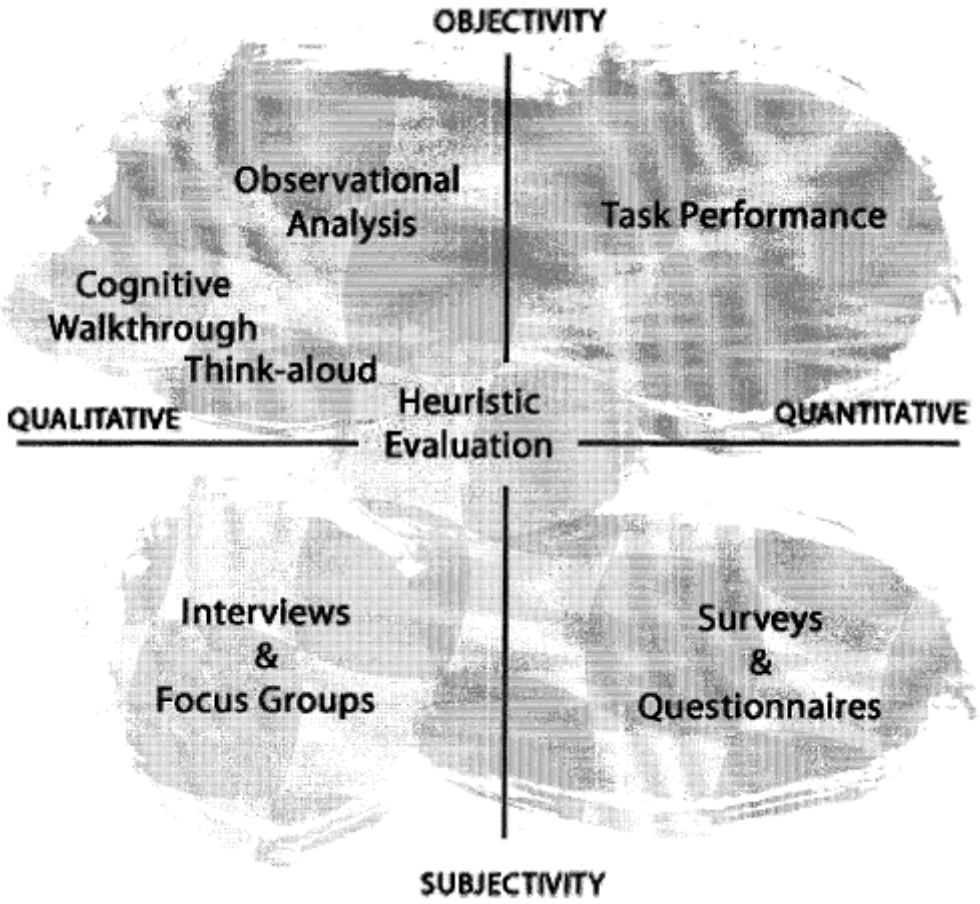

Figure 2.1: Current methods for evaluating productivity applications by Regan L. Mandryk [Man08]
While there are many different methods, they have varying degrees of objectivity and produce differing amounts of meaningful data.

behaviours based on this data.[PL99] There are, of course, other alternatives to retrieving data and using it for prediction purposes, as can be seen in other works. For example, data mining videos in order to create sets of strategies for an agent in a real-time strategy game[WM09]. This allows a very high number of expert data, and facilitates the creation of prediction models for how the agent should act based on both his plan and the opponent's predicted plan.

As a final remark on model creation, it is useful to create not only models for individual subjects, but to also create models for specific "archetypes" of players. With this, we can create a number of "schemas" which we can then use as basis to create new models for other subjects, by using the distance from a player's model to the created schemas[ELB13] or by using the schemas themselves to see how a general type of individual would perform in certain situations.[LS08] This allows not only rapid additions of new subjects, but also facilitates testing and changing gameplay elements based on the performance of the schemas.

One of the most popular models[SDM09] for emotional categorization, the OCC (Orton, Core and Collins)[AO88] model attempts to represent emotions through reactions to our perspective of the world, which is divided in three categories: events, agents, and objects. Emotions are then considered as valence reactions to the perceived events. This model is complex, but can be simplified by collapsing our reactions into a reduced number of positive and negative affective reactions. There are numerous models that build upon the OCC[SM07][VPB+06], and several papers written on formalizations of this model and ways to specify agent behaviour according to this model.

Another model[ENYI00], called FLAME - Fuzzy Logic Adaptive Model of Emotions, had the goal of both producing emotions as well as simulating emotional intelligence, using fuzzy rules (and by extension, fuzzy logic) to model the emotional process. This proved to be an improvement over previous models, since fuzzy logic helped capturing the fuzzy nature of human emotions. Alas, they separated the cognitive and emotional processes, while the interaction between them is what makes the human mind as complex to model.

Besides these models, there are many others that provide a complete mechanism to handle emotional responses, such as the EMA model or the WASABI model. The EMA model[MG09] does not represent a character's personality, instead using Beliefs, Desires and Intentions to emulate the personality. This makes the model harder to work with on some implementations. In addition, it requires appraisal and re-appraisal procedures used to emulate the dynamics of human emotions, which can be computationally expensive to develop and configure both during design time and development time for a video game engine.

The WASABI model[BA08] simplifies a human's mood as a linear unidimensional variable, which is simple in both implementation and development, since it only uses one criteria to compare emotional states, but becomes unrealistic by making relationships between mood and emotions dimmer. As we've seen in previous models, human emotions are complex mechanisms, and while a simple model is desirable for us, it is important not to sacrifice too much believability, or the potential gains will be hampered.

The Big Five Factors model[Wig96] attempts to emulate a human's personality in a way possible to be represented by a computer, using a set of five factors (openness, conscientiousness, extraversion, agreeableness and neuroticism). What they mean can be seen in figure 2.2 These factors can then be used to create a parametrized version of a personality, where each of the five parameters impact the agent's choices.

The PAD temperament model [Meh96] represents an emotional state as a point in a three dimensional orthogonal scale, which are the following:

- Pleasure-Displeasure (which represents affective states, classifying it as positive or negative)
- Arousal-Nonarousal (which represents how much mental or physical activity is required)
- Dominance-Submissiveness (which represents the degree of control over the situation at hand)

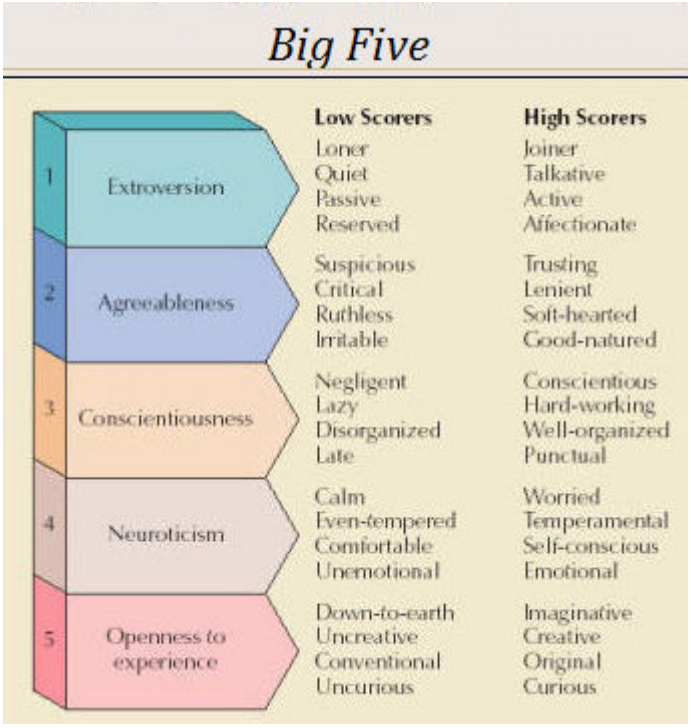

Figure 2.2: Big Five Factors. Representative illustration by Stuev["St11]

By arranging emotional states in this manner, it is possible to divide the emotional space into octants, which facilitates computation when determining where in the emotional state the subject is, and determine his behaviour based on that instead of more complex rules.

There are also model implementations, such as SHAME[PANK02], that are based on machine learning. This model is based on the OCC model[AO88] and introduces neural networks to learn how an emotional state is generated, using previously annotated data, as well as changes due to both internal and external stimuli. By using machine learning, they hope to model complex emotional phenomena in a modular way, making parts of the architecture domain independent and possible to be distributed, ensuring scalability, while keeping the implementation simple.

While more complex models can, in theory, produce better results, one must also take into consideration the performance of the model, since there is usually a time restriction on the results. Taking this into account, a simpler model is preferred, and, learning from previous works, fuzzy models are also preferred. A proposed model[NAR+14], based on mapping emotional states in arousal-valence dimensions[Rus80] estimated through their physiological readings, fits this need by using the variations in this emotional state on a regression-based approach.

Considering this approach, one can then extrapolate the players' reactions to different events and relate them with the variations in the emotional state, in order to determine how the agent should react to different events while using the created model to emulate the emotional state. Similarly to the PAD model, by using A/V values and mapping their emotional state into a two dimensional orthogonal scale, as illustrated in figure 2.3, we can achieve similar results with less

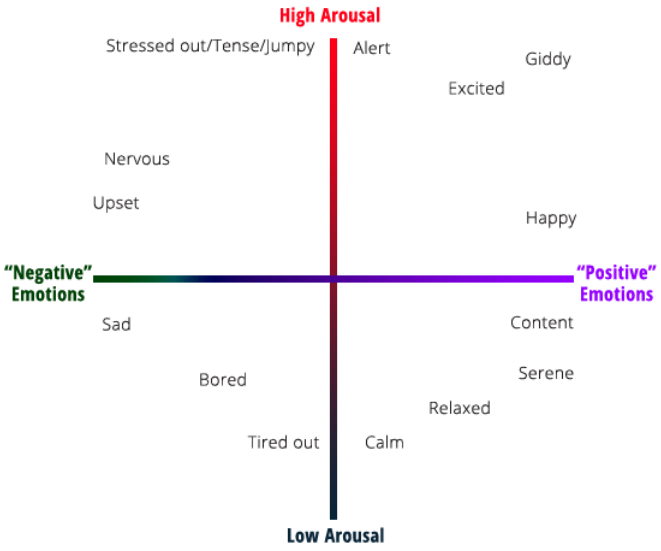

Figure 2.3: Arousal and Valence chart by Russel[Rus80]
Illustration by Isla[Isl]

computational resources required.

However, evaluating emotional models is still not a standardized affair, and each study shows his own results, normally through human interviews. There has been some effort in developing methods to evaluate these models, but it has not been widely adopted as of yet.[GM05] In our work, we have decided to compare both mechanical and subjective performance with that of human players.

2.2 Agents and Emotions

There are many alternative ways to define what an agent is [WJ95], but most of them agree on a few basic properties: the agent should be at least partially autonomous, or be able to control it's own actions without human intervention. It has to be able to react and perceive the environment in which he is inserted. Lastly, it can, but is not required to, communicate and interact with other agents.

Different approaches exist to construct agents, noted by different architectures. We can consider two main architecture types: deliberative and reactive.[Bro91]

Deliberative architectures contain an explicit model of the world and the agent makes decisions via logical reasoning. This logic is, of course, world/context dependent. As it stands, this architectures raise two issues: how to represent the world in an accurate and timely manner to an agent and how to make the information in that representation be representative of the knowledge and properties of complex entities must be such that it becomes possible to process and act upon it

in real time. An example of this architecture are BDI systems [RG95], in which an agent is viewed as having three mental attitudes of belief(B), desire(D), and intention(I).

Reactive architectures dismiss the usage of a symbolic representation of the world. The key aspects of these architectures are:

- Situatedness - Agent does not have to deal with an abstract world
- Embodiment - Feedback to the agents' actions is immediate.
- Intelligence - Certain factors influence actions, not just computational engine.
- Emergence - Intelligence comes from interacting with the world.

One should also consider hybrid architectures, which contain parts of both the aforementioned architectures as subsystems, normally by the use of "layers" which deal with information with different degrees of abstraction, such as a deliberative system controlling the high-level goals of an agent while having a reactive system handle the moment-to-moment interactions.

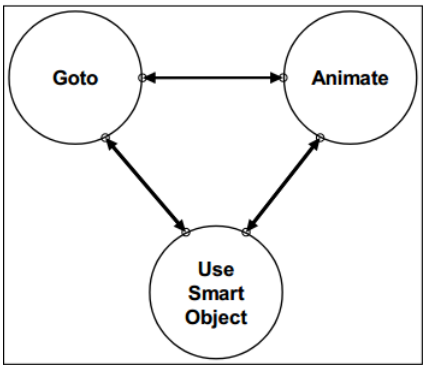

Figure 2.4: F.E.A.R. State Machine by Orkin [Ork06]

When it comes to video games, a common method to build agents is to use FSM and HFSM. These allow a network of states and transitions between those states which represent the action (or sequences of actions) the agent should take to reach another state. One can see that as the number of states increase, so do the number of transitions and system complexity. Behaviour in these systems tends to be robotic, since there's only a preset amount of transitions between states[HYK10]. A very "simple" state machine can be seen in figure 2.4, in that the simplicity is only at high level: one of the possible states is very complex.

Another method to build agents is to use BTs[Ale10], which present functionalities very similar to FSMs, but trade away some of the low level control on state transitions by focusing on the flow of decisions from the AI[Chr14]. This makes them a lot faster to develop, since we can reuse parts of the tree among the different agents, and we no longer have to consider every possible transition between states, since it's possible to execute multiple nodes at the same time.

Emotional agents attempt to emulate human behaviour using a symbolic approach[MZ07], using a symbolic emotional rule-base system with continuous interactions with the environment.

Despite their emotions being artificial, they can still be used to influence decision-making in varying ways. Comparisons of the performance of emotional agents with non-emotional agents is usually favourable. An example of an emotional architecture is EBDI (Emotional Belief-Desire-Intention)[JVH07] architecture, which attempts to use emotional data to influence the logical reasoning mechanisms, implementing them separately to allow different emotional models to be used, as seen in figure 2.5.

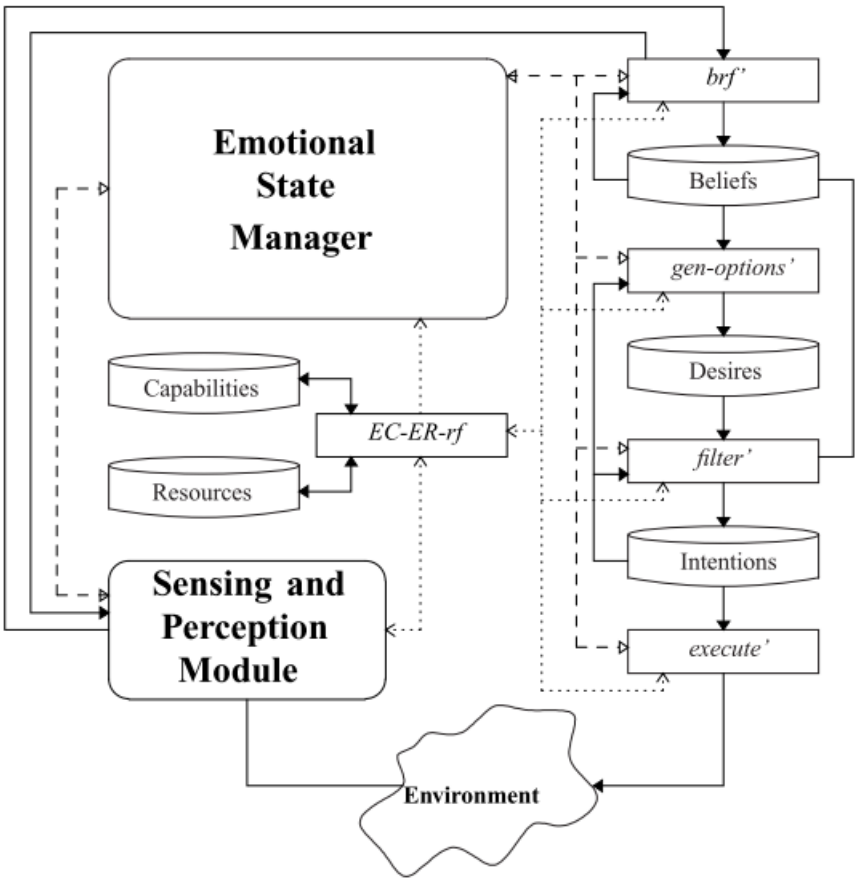

Figure 2.5: EBDI architecture [POMS05]
Compared to a typical BDI architecture, an EBDI architecture uses an emotional state to influence the agent's decision making proccess on all levels (Beliefs, desires and Intentions), which is in turn influenced by the agent's actions.

One particular implementation of an EBDI architecture attempts to introduce the concepts of Appraisal and Coping[GM04], which are based on a previous theoretical model[Laz94], which " organizes behaviour around two basic processes, appraisal (which characterizes the person's relationship with their environment), and coping (which suggests strategies for altering or maintaining this relationship)". There is then a third process, Cognition, which informs both of those processes, either by constructing mental representations that connect events to internal dispositions, namely the Desires in an EBDI architecture, or by suggesting and exploring strategies for

handling the person-environment relationship. One key feature we can take away from this work is that, as they state, " appraisal should not be construed as a deliberate process in itself, but rather a reflexive assessment of the current mental state, which may or may not have been elaborated by deliberation". This means we can, and should focus on the reactive component of our agent, when attempting to create an emotional agent.

An important notion in emotional agents is emotional decay[OS02]. If our goal is to emulate human behaviour as closely as possible, we must be aware that the emotional response is not static throughout time. It takes a certain amount of time to perceive stimuli and react in accordance to it (Reaction Time) and the response to it is not binary, as seen in figure 2.6, representing the emotional response to a given stimulus. Emulating this behaviour is crucial to making the agent as similar as possible to the human decision process.

Figure 2.6: Emotional Decay by Oliveira and Sarmento [OS02]
While emotional responses have varying intensities, they are not constant and the intensity of these reactions decays over time

It should be noted that, while there are contradicting studies regarding the effectiveness of these models and the emotional agents, there ares no studies that state that emotional agents hinder the interaction between agents and humans[BC09]. From this we can conclude that in terms of performance an emotional agent should at least perform similarly to a non-emotional agent.

Video games present a unique challenge for these types of agents, since they cannot allocate as many computational resources to the behaviour of NPCs[TM06]. In addition, it is important to remember that we not only have to predict how each agent will act, but we also have to make the model associated with that agent perform the relevant action, such as moving or other context specific action. This means that emotional agents, which are by definition more complex, have to work with simplified models that are adapted in order to accommodate specific game actions that could impact his behaviour. Despite these limitations, the usefulness of this type of agents for games is shown in a number of other works:

By using an affect-sensitive agent on an entity such as a robot, it is possible to create a better experience through empathic behaviour from the robot. The robot would change his facial expressions based on the performance of the child he played against, showing a positive reaction when the child played poorly and a negative reaction when the child plays well. In order to increase believability, they also developed a system to recognize affect to consider the child's emotions as well, making the agent appear more relatable when compared to one without such

a system[CLP+13]. In doing this, they prevented the robot's social presence from decaying over time, affecting the child's relationship with the robot in a positive way.

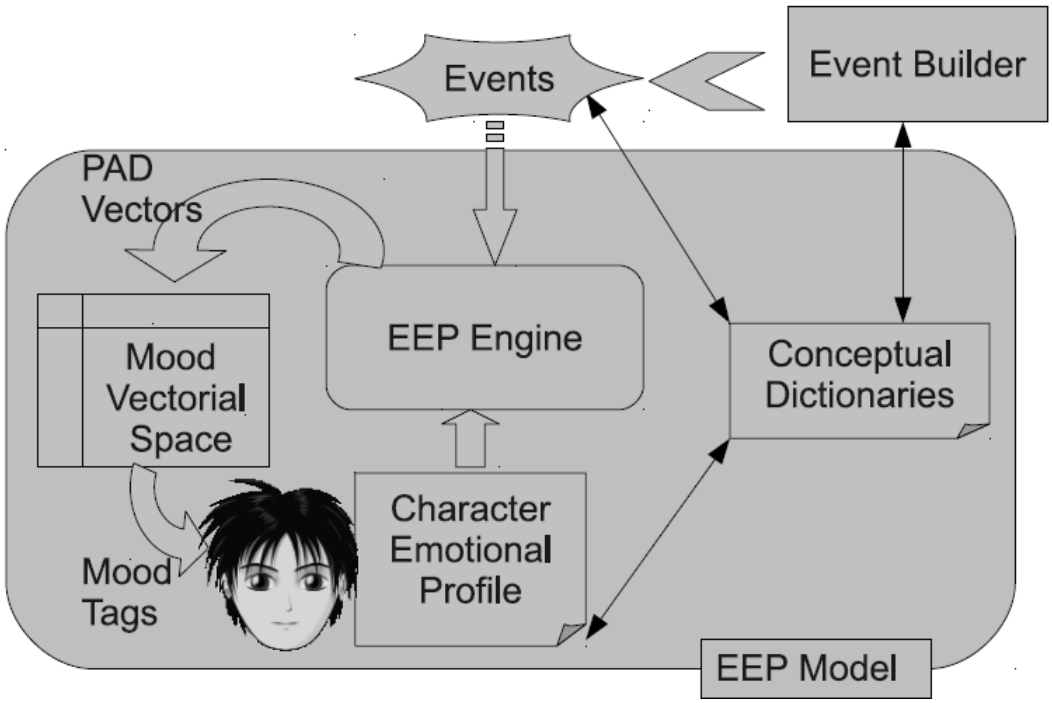

Figure 2.7: EEP General architecture by Luís Peña et al. [PnOPnS11]

Building upon not only the OCC model but also the PAD Temperament Model, the Emotional Elicitation Process (EEP) model[PnOPnS11] uses a simplified representation of the OCC model, applied to the dimensions of the PAD Temperament model, in an effort to create a robust yet computationally simple model for creating personalities for AI agents. They then tested the agent on a commercial video game (NeverWinterNights, Bioware) in order to compare how agents with emotional states compared to the agents seen during the game, by surveying a group of expert users. The architecture of this system is shown in figure 2.7, albeit at a high level.For the scenario they proposed, 13 out of 15 (86.67%) experts agreed that the agents were emotionally coherent and a significant improvement to the believability of their actions in game.

There have also been some efforts to create educational games by using affective NPCs, such as {ORIENT} (Overcoming Refugee Integration with Empathic Novel Technology), which also had positive results on the representation of agents as believable entities, with their own customs and able to generate feelings of empathy on humans. They used the FAtiMA EBDI cognitive appraisal-based architecture, which is based on the OCC model, as well as the PSI drives based model, based on the Big Five Factors model, which is a self-learning model that requires very little initial set-up, but also has a disadvantage of giving the developers less control over the actions an agent takes[LDAP12]. This architecture is represented in figure 2.8.

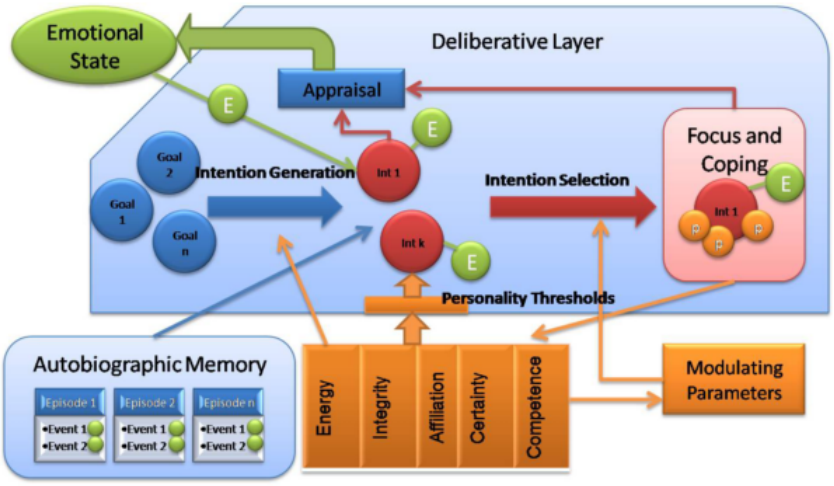

Figure 2.8: FAtiMA - PSI architecture by Mei Yii Lim et al.[LDAP12]

There has also been some efforts on making the game's narrative respond to the player's emotional state, and change accordingly[GPCC12], or actually generate content based on the player preferences[PTY10]. The implications of this work are that it is then possible to not only affect the story, but most if not all the aspects of a game based on the recovered and interpreted physiological dat [NRON13], or create more accurate simulators. This is even more relevant in Virtual Reality applications, where immersion is even more important, in order to create a better feeling of belonging in the virtual game world.

Chapter 3

Implementation

Our goal with this thesis is to create an agent which can emulate human behaviour, based on their emotional responses, in an attempt to use that data to improve quality assurance processes.

Validating agent-based models is still an issue[XLNV12], and while various methodologies have been proposed, there are none that are widely adopted, due to the amount of data needed to compare with the model outputs. A key issue is that data collection is difficult, since the studied natural systems are usually complex, especially as more agents are in the systems, such as crowd simulations. One proposed idea for validation is to adopt ideas from Human Computation to perform a task repetitively, taking the form of a game. By using this approach, which in theory is more appealing to human subjects, we can frame behavioural questions and attempt to capture the subjects natural and instinctive decision.

Drawing from this, one of the ways to validate our agent will consist on a simplified Turing test: we can compare the performance of the developed agent with the performance of humans by recording gameplay videos of both and running a survey to attempt to classify videos as either agent or human behaviour.

As with other AI agent works, it becomes paramount to build a simulator[Wal03]. This allows us to both get objective data, such as duration and preferences of actions, as well as enabling humans to interact with it, which will be used for the aforementioned simplified Turing test.

Based on our needs, we have decided to use the Unity game engine coupled with the RAIN AI engine, taking advantage of the asset store and the large community around those two engines to speed up development.

We will now delve into how each component was implemented with our chosen tools. To simplify, we will refer to the player or main agent as "player", him and his allies as "survivors",

while referring to the enemies as "zombies".

3.1 Simulator

It was necessary for our simulator to be representative of the game used to retrieve our initial data. This means that our simulator has to have basic FPS mechanics (moving, aiming and firing), and comparable zombies. We will now go further in depth on each component developed.

3.1.1 Map

The first step was creating a suitable area for the agents to act. Based on our initial game, it was important to have a map that a mix of different environments, to showcase how the agents behaved with varying degrees of proximity. We have created a single map, which can be seen in figure 3.1, which we believe has the required elements.

Figure 3.1: Simulator Map

The end goal of the map is to reach the room furthest to the top-right of the map, starting at the room furthest to the left possible.

Throughout the map, there are some obstacles, in the form of walls and other objects, such as boxes and machinery. These are used to make the map interesting for human players and to give some tactical options to the player.

3.1.2 Communication

Agents can communicate between themselves. This works by having the agents subscribe to a specific communication channel, which differs according to whether or not the agent is friendly or hostile to the player. After subscription is done, whenever an event worth reporting happens,

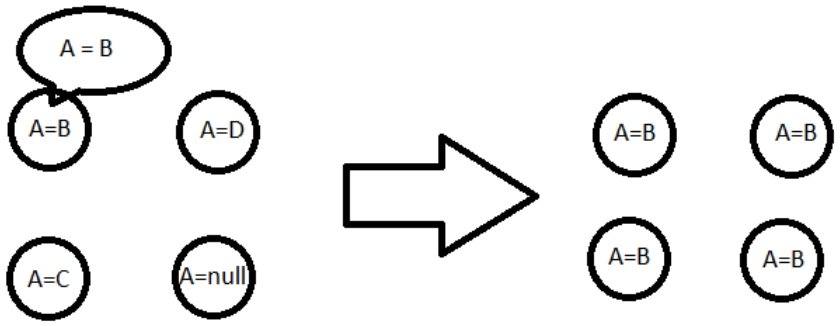

Figure 3.2: Broadcast example
Upon an order being broadcast, all subscribed agents replace their current variable with the broadcasted statement

it is broadcast through that channel. This makes a specific variable on the agents subscribed to that channel be set to the contents of the broadcast. An example of how this works can be seen in figure 3.2

Currently, the only events being broadcast are the discovery of a health pack for the survivors and the discovery of a survivor for the zombies.

3.1.3 Pickups

Scattered through the map are a number of pickups which assist the survivors in completing their goals. This aid is in the form of medical kits, which are used to restore the survivor's health. When a survivor spots one, its location is broadcast to all survivors, but only the player may pick it up and use it. The player can have any number of medical kits in his inventory. Using a medical kit restores up to 75 health to a survivor, limited by their maximum health. Since player health has a direct impact on his A/V values, it is important to have a way to recover health, similar to the one used in the base game.

3.1.4 Waypoints

There are a number of waypoints across the map. They are used for two different purposes: on one hand, they are used to mark the general path the agent should take. However, they are also used to mark certain locations as possible flank paths or cover spots. These other types are used by the agents to determine more favourable positioning when in combat. Waypoints in general are used to simulate a player's knowledge of the map. In figure 3.3, we can see the default path the agent will take which allows him to reach the end of the level.

3.1.4.1 Flanks

Flanks are connected waypoints that mark specific paths the agents should take when engaging the enemies. These differ from the normal waypoints in the sense that they do not mark a direct path

Figure 3.3: Path to the end of the map
Default path used by the agent to simulate prior knowledge of the game world

through the level, but a series of points that mark alternate paths to reach a given area.

3.1.4.2 Cover

Cover points are disconnected waypoints that mark specific locations where the agent has a positional advantage in the environment. These are usually points near walls or behind obstacles, so that the survivors can position themselves better when in combat.

3.1.5 Weapons

There are three weapons implemented, representing the main three weapons of our base game. They are useful not only to differentiate players, by determining their favoured weapon, but also in order to keep the player as close as possible to the conditions present in the base game.

The weapons have the following characteristics:

- Assault Rifle - Full Automatic fire rate, average accuracy
- Rifle - Semi Automatic fire rate, higher accuracy
- Shotgun - Semi Automatic fire rate, bullet spread

The player has the three weapons on his inventory. While each weapon has a different magazine size, which forces the survivor to reload, they have unlimited ammunition. Due to this, it is possible to better determine what is the favoured weapon of the player, since he won't run out of ammunition for it.

3.1.6 Commands

One aspect that was helpful was allowing a way to control, even if not directly, some aspects of the agent decision making process. This is done by allowing certain keys to edit some of the agents properties. The possible commands are:

- Enable/Disable Fire
- Enable/Disable Retreat
- Enable/Disable Swap Weapon
- Swap Agent A/V Model

This last command is used to cycle through the various agent models developed, which could be useful if coupled with features such as save states, allowing fast and repeated testing of a specific portion of the simulation without having to restart it.

3.1.7 Allies

Our base game has a cooperative aspect, by allowing up to four players to combine their efforts in order to achieve the success conditions of the levels, but also allows a single player to attempt this, by giving it computer controlled allies. These aren't as smart as a human player, and so, in order to replicate them in our simulator, do not require as complex of a behaviour when compared to the player. They are simply there to assist the player in reaching the end of the map, while having no emotional component of their own.

- Once they spot an zombie, if the player is not avoiding combat, they will initiate combat and shoot against it.
- If they spot a pick up, they will broadcast its location.
- If in combat, they will attempt to follow the Flank and Cover markers to the best of their ability, not straying too far from the player.
- Otherwise, follow the player.

3.1.8 Enemies

Enemies are one of the most important elements of Shooter games, and in our base game, they took the form of zombies, who wandered around the map. In order to represent these enemies, our simulator has a number of zombies with a simplistic behaviour, which serve as obstacles to hinder the player from achieving his goal. Their behaviour is as follows:

- Wander Randomly

- If they spot a survivor, alert nearby zombies (within 10 meters, blocked by terrain) and charge at the survivor.

Zombie attacks remove health from the survivors. If a survivor reaches 0 health, it dies. If the player dies, the simulation ends.

3.1.9 Statistics

One very important required feature is a way to track certain events in the engine for statistical purposes. Since our goal is to compare performances between agents and human players, it is important to have an objective look at how they behave, so we have more data to compare with.

Currently, the simulator tracks the following statistics:

- Favoured Weapon - The weapon used for the longest time
- Survived - Did the player survive through the simulation?
- Time in Zone 1 through 4 - How long did the player take in each section
- Total Time Taken - How long did the player take through the simulation
- Accuracy - Percentage of shots hit by the player
- Zombies Killed - How many zombies the player himself killed.
- Health Lost
- Allies Dead

These are written in a file after the simulation ends, either through reaching the end of the map or by dying, and are kept in another entity than the player. When the player experiences a relevant event, it signals that entity so it can log it correctly. There are 5 events being tracked: entering a zone, firing a weapon, hitting with a shot, killing a zombie and losing health. The other statistics are captured at the end of the simulation by the entity itself. These signals are done by direct function call, since the entity is static in the engine.

3.2 Agent

The AI developed needed to not only have a preset plan to complete the simulation, but mainly a reactive component that could quickly respond to the events triggered by the agent interaction with the simulator. This means that we needed a simple plan and a set of rules that could quickly change the agents actions. As such, we have opted for a Behaviour Tree model.

BTs[Chr14] are a tree of hierarchical nodes which control the flow of decision making of an AI entity. The leaves of the tree are actual commands that control the AI, while the other nodes are utility nodes that control how the AI travels through the tree to reach the set of leaves most

suitable to its situation. Trees have varying degrees of depth, which is important, since we can implement simple behaviours in small trees which can then be used on multiple agents, forming a sort behaviour library which can be reused in different situations.

Due to how a tree is organized, it is simple to design a high-level behaviour by chaining simpler behaviours ordered by their desirability, representing a sort of "fallback tactic" when more desirable behaviours fail. Managing how the tree is kept and optimizing its behaviour is not within the scope of this thesis, and as such is kept entirely by the Rain AI agent.

During runtime, nodes have three relevant status:

- Success - The operation on this node was a success.
- Failure - The operation of this node was a failure, or it is inactive.
- Running - The node is still running.

This means a node may be active for an indefinite amount of time. This happens more frequently in utility nodes, which we can split in two types.

Composite nodes are any node with more than one children. Depending on the node, it will execute these children on a given sequence, succeeding only when all children succeed and failing if one children fails, being in Running status through the rest of the execution.

Decorator nodes only have one single child. They typically are used to transform the results received, but can also be used to terminate nodes after a given time has elapsed, or, as used in our agent, to repeat processing of its child, in order to force the agent arousal/valence values to update regularly.

Behaviour wise, our agent is quite simple. We'll start by showing how it works, and then we'll discuss the emotional model attached to it.

3.2.1 Behaviour

Due to using a BT, it's fairly simple to showcase the high-level behaviour of an agent by using a list. The further up the list the behaviour is, the higher the priority of said behaviour.

- IF zombie spotted, decide between:
  - Enter Combat, Randomly Order to Take Cover or Flank enemy, Fire at Enemy Until Dead.
    - IF Enemy at long range AND Number of enemies Low, use Rifle
    - IF Enemy at medium range, use Assault Rifle
    - IF Enemy at short range AND Number of enemies High, use Shotgun
  - Avoid Combat, Keep distance to zombie, Success.[1]

[1] This makes it continue to follow the waypoints, as a fallback behaviour, while still keeping a set distance to the zombie.

  - Retreat from Combat, moving away from the zombie as fast as possible. When distant enough, Success.[2]

- IF Medical Kit detected, Retrieve it (Move to its location).
- IF Low on Health, Use a Medical Kit to restore health.
- IF Ally Low on Health, Use a Medical Kit to restore ally's health
- Follow Waypoints to end of map.

This list also shows the variety of actions that the agent has in the world. We can reduce them to "Move to location", "Fire Weapon" and "Change Weapon". Each agent also has a favoured weapon. This fact changes the thresholds of enemies required to trigger a weapon change. The goal of the agent is to reach a location at the end of the map.

3.2.1.1 Knowledge

The agent is aware of its own status (health values, ammunition still on the weapon, A/V values), his allies current health, in addition to the location of every waypoint (be it his path or the flank and cover waypoints) in the map. He is also aware of the locations of medical kits any of the survivors have spotted.

3.2.1.2 Sensors

To aid in decision making, the agent has three visual sensors to represent human sight. This assists us in determining distance by simply checking in what sensor it is visible, be it the close range, medium range or long range sensor. These sensors are based on Raycasting and are manually polled every time the tree is analysed.

3.2.2 Data

The data we're using to create our models comes from actual game playthroughs of a popular shooter game, Left4Dead 2, and gives us the Arousal and Valence values for a given timestamp, as well as the current event, which can be seen in figure 3.4. Each captured sample represents 10 seconds of playthrough. We then convert that data to a more suitable format for our use, by storing only the initial A/V states, retrieved by averaging the data for the initial 2 seconds, and their highest variation during the remaining 8 seconds, as well as the subject and event names.

We then converted this data into a linear regression model for each event for each subject, creating a regression surface unique to every subject. This allows us to compare the created surfaces between subjects and create an euclidean distance matrix, represented in figure 3.5.

By making a distance matrix between every pair of subjects we can cluster them, which will create a number of clusters that will serve as "archetypes", which will serve as a basis to the

[2] See footnote 1

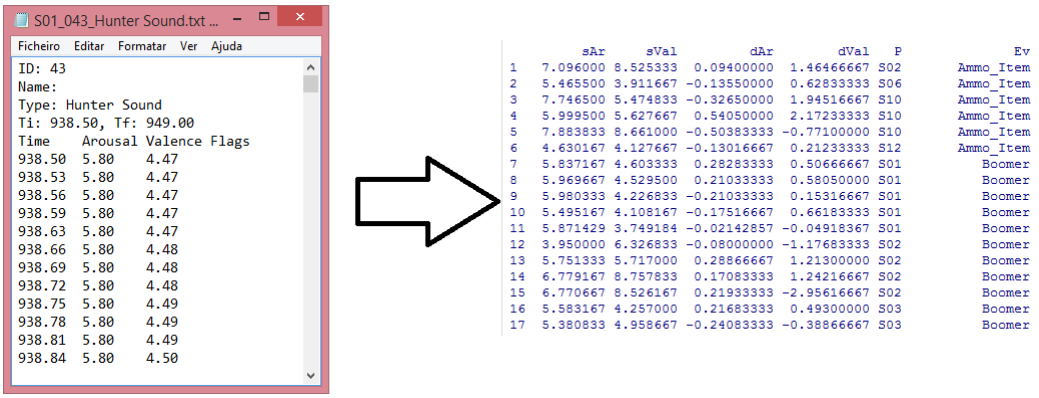

Figure 3.4: Data gathered from Left4Dead 2 and conversion
Data gathered from multiple Left4Dead 2 playthroughs, divided by subject and event. We then convert this data to a simpler format, keeping only the starting A/V values, the maximum variation, as well as the observed subject and event.

behaviour model we must build. The obtained clusters can be seen in figure 3.6. We then create models for each of the obtained clusters, by considering the data we have on every player in that cluster.

> "Having these models enables us to determine where each subject is located compared to each cluster, aiding us in the process of determining the subject's behaviour: we do this by creating a fuzzy membership vector, which represents the relative distance of a subject to each cluster model, allowing us to have a certain degree of uncertainty when predicting reactions while also differentiating between members of the same cluster. With this, we can add new players without rebuilding the created models." [NAR+14]

In addition, we also have the gameplay videos of the playthroughs. This allows us to gather additional data, such as the player's favoured weapon, that we then incorporate in our models, as well as enabling us to determine the set of actions the player tends to take in those situations.

3.2.3 Models

Each subject is then composed of the following data:

- Favoured Weapon
- Matrix of A/V Reactions to Events

With this data, we can extrapolate the kind of responses the player will have to the simulator.

On each cycle, in parallel with the above BT, we update a counter in order to update our A/V values. After a certain amount of cycles, roughly taking 0.25 seconds in an attempt to match the

	S01	S02	S03	S06	S10	S12
S01	0.000000	2.41188802	5.636891	2.43092823	70.48187	8.462151
S02	2.411888	0.00000000	8.048779	0.05394563	68.06998	6.050263
S03	5.636891	8.04877868	0.000000	8.06781889	76.11876	14.099042
S06	2.430928	0.05394563	8.067819	0.00000000	68.05094	6.031223
S10	70.481869	68.06998073	76.118759	68.05094052	0.00000	62.019718
S12	8.462151	6.05026299	14.099042	6.03122278	62.01972	0.000000

Figure 3.5: Distance Matrix
Matrix created by calculating the distance between each subject's emotional responses.s

average human reaction time[Hum], the agent's inner A/V values are updated based on the current simulator properties and the agents significance multipliers.

However, the agent has two sets of A/V values. We'll call one of them the Effective values and the other set Decay values. Every time we update the A/V values of our agent, it immediately updates the decay values. These values are calculated as follows, split into two equations for clarity.

$$Arousal = 5 + \frac{1}{2}(S_{A1} * (Z_S + Z_C)) + S_{A2} * V + \frac{1}{H} * H_{20\%} + S_{A4} * D \tag{3.1}$$

$$Valence = 5 - \frac{1}{2}(S_{V1} * (Z_S - 3) + S_{V2} * (Z_C - 3)) - S_{V3} * V + S_{V4} * H - S_{V5} * D + S_{V6} * MK \tag{3.2}$$

Followed by

$$Arousal/Valence = \frac{Arousal/Valence - 5}{5} \tag{3.3}$$

Where Zs is the number of zombies in sight, Zc is the number of zombies in close range, which includes the number of zombies in sight, V stands for the number of volumes within 3 meters, to simulate a feeling of claustrophobia, while excluding the zombies themselves. H and H20% represent the current health percentage and if the agent is under 20% health, respectively. D is the number of dead allies, MK stands for the number of Medical Kits in the inventory and Sx represent the Subject's Emotional Multipliers, which go from 0 to 1. Sax may differ from Svx. We then adjust the values to fit into a -1 to 1 scale, instead of a 0 to 10 scale, to use in other aspects of the agent, such as accuracy. High arousal lowers accuracy depending on the weapon used.

After this adjustment is done, we then update the effective A/V values in the following way:

$$EffectiveA/V = \frac{3 * DecayA/V + EffectiveA/V}{4} \tag{3.4}$$

This allows the Decay values to fluctuate rapidly as the simulation advances, while the Effective values rapidly tend towards them over time, which helps simulating response times and

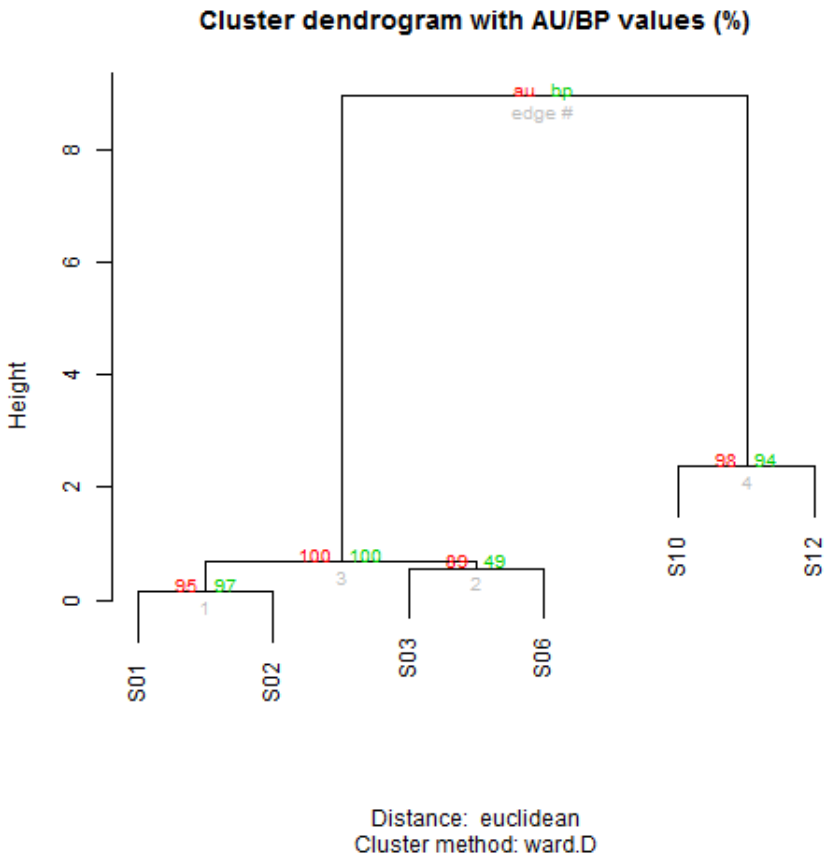

Figure 3.6: Clustering Results
Taking into account the distances calculated previously, we can group our subjects into "archetypes"

instinctive reactions that happen before a person recognizes what is happening.

Finally, on the topic of A/V variations, killing a zombie directly modifies the Effective Arousal by .1 and increases the Effective Valence by .05. This is to represent the general feeling of relief and satisfaction of removing a threat. We used the term modifies because the change is applied in the direction of 0, so if a player is at .56 Effective Arousal, killing a zombie decreases this value, while if he was at -.56 Effective Arousal, it would increase this value.

As a rule, the multipliers are lower on the arousal scale, since most of the time they are increasing the arousal value of the player. The arousal value also tends to be elevated. A/V values past certain ranges, as determined by our analysis of the clusters, enable or disable certain behaviours, like Entering Combat or Retreating. This can be overridden with the command system mentioned previously, but performing those actions will still impact the agents A/V values.

Additionally, Effective Arousal also has an impact in accuracy. Depending on the weapon used, high arousal lowers accuracy by different amounts, with the rifle suffering the least impact of all the weapons.

Chapter 4

Results

In order to validate our solution, we performed two different experiments in order to retrieve meaningful data. We will review and discuss the data collected from both of these experiments.

4.1 Experiment Description

The first experiment consisted on having a number of people play in the simulator, taking the role of the player, in order to have statistical data on humans, as described in 3.1.9 . It was allowed for the same person to repeat the experiment, with the results of their attempts being averaged before compared. We then had the AI run the same simulation, and averaged the result of their runs. All runs had the same map and starting situation, as well as the same number of enemies, but since some of the behaviour of the other agents is random, it influenced the behaviour of the player as well. This is acceptable for us.

The second experiment consists on a simplified Turing test in the form of a survey: we capture short videos[1] of certain gameplay sessions, to later show to other individuals. These individuals first have a brief hands-on experience in the simulator, so they can better understand what they are about to see. Amongst all of the videos, there are three types: we have 1 control video, which is always shown first, which is clearly an AI agent playing through the simulator. This is used to identify if the surveyed individual is taking it somewhat seriously, and if their answers are reliable. The other two video types are AI videos and Human videos. The goal of the survey is to determine if an individual looking at those videos can identify if the video they are watching is of a Human gameplay session or an AI gameplay session. We have 6 total videos (excluding the control video), 4 of these are Agent videos and 2 are Human videos. After the survey, we will build a confusion matrix with the results in order to determine if the agent was successful in emulating human behaviour or not.

[1] Between 10 to 20 seconds

Both these experiments were run/recorded on the following machine, which was also the primary development environment:

- Windows 8.1
- 2.4GHz i7 processor (8 virtual threads, 4 physical cores)
- 16GB of Ram
- Unity3D Engine v4.6.2f1, without deploying the project
- RAIN AI Engine v2.1.8

4.2 Results

We will now analyse the results obtained through our experiments.

4.2.1 Statistical Data from the Simulator

The first test was run for a total of 10 human subjects, each performing multiple playthroughs on the simulator (from 2 to 4), resulting in a total of 34 results. The agents were run a higher number of times, for a total of 38 playthroughs. Some of the agent subjects required player intervention to force certain actions in order to advance, which we will discuss later. We will reproduce the results of this test in tables 4.1 and 4.2.

Table 4.1: Simulator Statistics

	Minimum		Average		Median		Maximum	
	Human	Agent	Human	Agent	Human	Agent	Human	Agent
Time in Zone 1 (s)	14	39	43,56	50,00	44	47	65	100
Time in Zone 2 (s)	0	43	51,67	64,06	52	55	90	118
Time in Zone 3 (s)	0	0	39,52	40,18	44	46	72	60
Time in Zone 4 (s)	0	0	33,07	33,41	40	42	63	56
Total Time Taken	35	138	167,81	187,65	174	183	286	238
Accuracy	11%	55%	62%	77%	63%	78%	100%	95%
Kills	1	6	20,15	25,24	18	25	54	47
Health Lost	0	0	58,15	35,88	40	10	170	140
Allies Dead	0	0	1,04	0,94	0	0	3	3

Table 4.2: Favoured Weapons

	Human	Agent
Assault Rifle	33%	29%
Shotgun	56%	6%
Rifle	11%	65%

Lastly, the agents had a measured success rate of 76%, that is, they reached the end of the level 76% of the time, while human players had a slightly lower success rate at 74%. This is not enough of a difference to be significant, and is a positive sign that they are this close. The number of dead allies proved a mostly irrelevant statistic, since it was similar for both agents and humans, and not a differentiation criteria.

Looking at the times taken in each zone, the minimum value is mostly filled with zeroes. This means the subject did not even reach that zone. As such, it is to be expected that when the total time taken is at its lowest, the subject did not reach the end of the level, which is true in both the agent and human scenarios. Despite the significantly higher minimum total time, it is important to note that the agent did not cover that much ground. Those results happen when the agent is too scared to move forward, and as such, takes a much longer time making progress (or requires an player input to force him to advance), not that it actually reaches farther into the simulation compared to the human.

The average and median values for the time taken in each zone, however, are much closer. While the times in zone 3 and zone 4 are lowered by the higher number of zero times, due to player death, one can conclude the agents take a similar amount of time in each zone compared to humans. On zones 1 and 2, which aren't as open as the latter zones, it's harder for the agent to avoid conflicts, which result in longer zone times whenever he attempts to do such an action and sees himself forced to engage the enemy or to retreat.

Maximum times are higher on the human, when we look at all the zones together, but, for the same reason we previously stated, on earlier zones the agent might see himself with no viable paths that he is willing to take, and cowers on the edge of the enemies sight while attempting to find an opening. The humans take longer not due to lack of skill, but due to the desire to explore: While the agent is aware of the simulation map, the human is not as familiar with it, and often went to explore different paths, in the hopes that he could find a medical kit.

Accuracy numbers are far higher for the agent, but a dedicated human player can (and did) accomplish perfect accuracy, at the cost of time taken. However, since our playthroughs are from different people, with different familiarities with this type of game, it is to be expected that the deviation in accuracies is higher on humans. From observing the subjects run the simulation, it was clear that not all of the subjects took it seriously, and were not doing their best. The same conclusion can be taken when we look at the *Health Lost*, but that is also due to the agent's favoured weapon.

Analysing the number of kills done by the agent, we can see it is almost 25% higher than the number of kills done by humans. This stems from two facts:

- Due to the targeting heuristic, the agents favour the Rifle, which not only is more accurate at longer distances, but also deals more damage. Perhaps even more relevant is the fact that the rifle target acquisition range is larger than the allies' target acquisition range, which means that they will not shoot at that enemy and take the kill for themselves.

- The agent is much more precise when firing. Since he does not miss as many shots, there is a lesser chance of his allies killing the enemies for him.

It's relevant to note that a dedicated player can, and will, kill more enemies than the agents if he sets that as his goal. However this, again, comes at a cost in both time and health.

4.2.2 Simplified Turing Test

The second test was run for a total of 41 human subjects, each of them viewing a series of 7 short videos captured in the simulator through the use of an external tool, OBS (Open Broadcaster Software), recording gameplay from a third person perspective. These videos represented specific situations, and the enemy positions were adjusted in order to better showcase the agent's response in a believable way.

As stated in the previous section, one of the videos was a control video, in which it was as obvious as possible that the player was being controlled by an AI. This was done by using the default shooter AI we acquired from the sample project used to showcase the RAIN AI engine, which was sufficient to create this control video. After identifying this video, we would then ask the subjects to examine a series of videos and determine which were controlled by agents, and which were controlled by humans. We did not reveal the percentage of videos of each type, or any other information about the videos, other than what they would be watching compared to their experience seeing the simulator first-hand. The videos were shown in a randomized order, except for the control video, which was always first, in order to prevent guesses based on the number of humans and agents guessed previously and the videos remaining to inflate a particular video's score. We then tallied the results and created the following tables:

Table 4.3: Survey results

	Is Agent		Is Human	
Agent Guess	114	69,51%	12	14,63%
Human Guess	50	30,49%	70	85,37%

Table 4.4: Calculated Parameters

True Positive Rate, Sensitivity	69,51%
False Negative Rate, Miss Rate	30,49%
False Positive Rate, Fallout	14,63%
True Negative Rate, Specificity	85,37%
Positive Predictive Value, Precision	90,48%

Before we analyse the results, we would like to make the following observations:

- Humans are very good at identifying other humans, but can still sometimes make mistakes. This was seen specifically in a single group of subjects who took the test together, and

seemed to base their own guesses off the guesses of the others, especially if said subject was particularly vocal of his opinion. We believe this may have been a mistake on our part to allow the group to participate together and count each individual answer, instead of grouping their answers into a single "subject", since their answers matched. If we did so, there would only be a single misclassification on the humans, which is more in line with our expectations.

- Despite our best attempts, some subjects had difficulties understanding what they had to pay attention to. We believe we have signalled and excluded the answers of those subjects in this analysis, but it is possible their guesses have influenced others who took the survey near them.

Taking into account our observations and our goal, out of all the data retrieved, the most important aspect is the miss rate, which is a little over 30%. This means that roughly 30% of the agent videos were misclassified as human videos, which is a very hopeful number. Modelling human behaviour is difficult and since our simulator is not an exact replica of our source game, it was expected that our model wouldn't be as accurate and wouldn't give high results. It should be noted that not all of the developed models reach this rate: we selected the model we determined as the most able, from our own observations, to create the simulation videos.

4.3 Results Discussion

From the first experiment, we can conclude that our agents have similar mechanical performance compared to a human person. A dedicated individual may outdo the agent baseline, but in general, the agents have a far stabler performance when compared to different people. Despite this, we consider it a success that the performance is not drastically worse than the normal human player, since it's easier to adjust the agents to become worse than it is to improve them. From this experiment alone, if it were not for the extra time taken when developing this type of agent, we would recommend the usage of emotional agents for game testing purposes, since they already give the developers some feedback on troublesome areas, based on where they spend much longer than normal.

With the second experiment, we conclude our agent managed to emulate human behaviour within our initial expectations, which due to the available time for the development of the simulator and initial data used was expected to be at most 40%, by generating almost one third of misclassifications. In order to improve this number, we would have to further develop the simulator or, alternatively, generate new models based on the current simulator. We believe that this process to create behaviour models is a step in the right direction to achieve our goals of emulating human behaviour, but will require more extensive testing in order to validate it completely. As a proof-of-concept, it has shown promise.

Chapter 5

Conclusions and Future Work

In this thesis we discussed the creation of an AI Agent that would be able to emulate a player's behaviour on a specific type of gamer based on their emotional state. We also discussed different methods to model a human's behaviour, and implemented one of those methods based on Arousal and Valence values derived from physiological data acquired in human playthroughs. Finally, we experimented with our proof-of-concept in order to prove its viability.

5.1 Goal Assessment

The simulator created was a satisfactory representation of our chosen game (Left4Dead 2), and implemented most of the main features of that game, except for the multiple enemy types. It currently only has one map, but it is relatively easy to add new maps: The agent only cares for a Navigation Mesh it can use to travel the map, a set of waypoints to give him direction, and some information on interesting tactical positions in order to improve his in-combat performance.

The agent emulates human behaviour, although not perfectly, and is capable of completing his goals in the simulation with similar performance to a human player. He would not pass a Turing test if being observed continuously through the simulation, noticeably due to his pre and in combat behaviour during certain engagements, but some moments are significantly harder to distinguish, especially when executing "cowardly" actions such as evading.

In general, we feel it is an asset to game developers, especially due to the information one can get by monitoring the emotional state variances of the agent as well as his "mechanical" performance. However, it is important to remember that creating models for the games being tested take time and a lot of a gameplay data in order for the models to be convincing, which is a strong deterrent for smaller scale games.

The usage of these agents also brings benefits in other areas: giving Non-Player Characters an emotional component would make their behaviour more realistic, which would help both in

immersion and connecting with the characters. From what we could gather, this usage of emotional agents has not yet been put into practice in commercial video games.

5.2 Future Work

There are various ways to progress from this work: On the simulator side, there is plenty of room for improvement, namely improving the animations and how they are used, creating new animations, different enemy types (to solicit different reactions from the player), usage of luminosity to impact the players emotional state (which was cut due to being resource intensive in the Unity version used in development), as well as the usage of sounds, both as immersion mechanism as well as another type of signal to track for the agents, improving detection and reaction mechanisms.

One could also develop a map builder/generator, in order to facilitate the creation of new maps, or change the data being tracked by the statistics module, in order to adapt it to different game types.

The agent is implemented based on Behaviour Trees, which makes it modular inherently. It is possible to reuse most of the developed behaviours and combine them differently, even for someone with low programming experience, thanks to the GUI. Developing new behaviours is also streamlined, with actual code only being needed when implementing new actions. Ideally, the agent would be using the same source for their models as the environment it will be evaluated in, so as to improve results. Additionally, the models are currently embedded into the agent, which is not the best way to implement, since it requires altering code to alter the models: it would be better if the agent could load those models during runtime, from external files, which facilitates changes in a rapid development environment.

It's also possible to combine the current agent architecture with machine learning mechanisms, in order to have the agent learn and improve over time, while possibly making the agent become slightly game agnostic, which is similar to Sentio characters developed by RivalTheory (RAIN AI engine developers).

References

[Ale10] David Hernandez Cerpa Alex Champandard, Michael Dawe. GDC Vault - Behavior Trees: Three Ways of Cultivating Strong AI, 2010.

[AO88] A. Collins A. Ortony, G.L. Clore. *The Cognitive Structure of Emotions*. Cambridge University Press, Cambridge, 1988.

[AS10] Apostolos Ampatzoglou and Ioannis Stamelos. Software engineering research for computer games: A systematic review. *Information and Software Technology*, 52(9):888–901, September 2010.

[BA08] Christian Becker-Asano. WASABI: Affect simulation for agents with believable interactivity. 319:186, 2008.

[BC09] Russell Beale and Chris Creed. Affective interaction: How emotional agents affect users. *International Journal of Human-Computer Studies*, 67(9):755–776, September 2009.

[Bro91] Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47(1–3):139 – 159, 1991.

[Chr14] Chris Simpson. Gamasutra: Chris Simpson's Blog - Behavior trees for AI: How they work, 2014.

[CLP+13] Ginevra Castellano, Iolanda Leite, André Pereira, Carlos Martinho, Ana Paiva, and Peter W. Mcowan. Multimodal Affect Modeling and Recognition for Empathic Robot Companions. *International Journal of Humanoid Robotics*, 10(01):1350010, 2013.

[ELB13] Marlon Etheredge, Ricardo Lopes, and Rafael Bidarra. A Generic Method For Classification Of Player Behavior. *Aiide*, pages 2–8, 2013.

[ENYI00] Magy Seif El-Nasr, John Yen, and Thomas R Ioerger. Flame—fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-agent systems*, 3(3):219–257, 2000.

[GM04] Jonathan Gratch and Stacy Marsella. A domain independent framework for modeling emotion. *Cognitive Systems Research*, 5(4):269??306, 2004.

[GM05] Jonathan Gratch and Stacy Marsella. Evaluating a computational model of emotion. *Autonomous Agents and Multi-Agent Systems*, 11(1):23–43, 2005.

[GPCC12] Stephen Gilroy, Julie Porteous, Fred Charles, and Marc Cavazza. Exploring passive user interaction for adaptive narratives. *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces IUI 12*, page 119, 2012.

REFERENCES

[Hum] Human Benchmark. Human Benchmark - Reaction Time Statistics.

[HYK10] Frederick William Poe Heckel, G Michael Youngblood, and Nikhil S Ketkar. Representational complexity of reactive agents. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 257–264. IEEE, 2010.

[Isl] Isla. Monitoring Emotions: Valence vs. Arousal - The Thinking Zygote.

[JVH07] Hong Jiang, Jose M Vidal, and Michael N Huhns. Ebdi: an architecture for emotional agents. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 11. ACM, 2007.

[Laz94] R.S. Lazarus. *Emotion and Adaptation*. Oxford University Press, 1994.

[LDAP12] Mei Yii Lim, João Dias, Ruth Aylett, and Ana Paiva. Creating adaptive affective autonomous NPCs. *Autonomous Agents and Multi-Agent Systems*, 24(2):287–311, 2012.

[LS08] Craig a. Lindley and Charlotte C. Sennersten. Game Play Schemas: From Player Analysis to Adaptive Game Mechanics. *International Journal of Computer Games Technology*, 2008:1–7, 2008.

[Man08] Regan L. Mandryk. A physiological approach for continuously modeling user emotion in interactive play environments. *Measuring Behavior 2008*, pages 93–94, 2008.

[Meh96] Albert Mehrabian. Analysis of the big-five personality factors in terms of the pad temperament model. *Australian Journal of Psychology*, 48(2):86–92, 1996.

[MG09] Stacy C. Marsella and Jonathan Gratch. EMA: A process model of appraisal dynamics. *Cognitive Systems Research*, 10(1):70–90, 2009.

[MZ07] Khulood Abu Maria and Raed Abu Zitar. Emotional agents: A modeling and an application. *Information and Software Technology*, 49(7):695–716, 2007.

[NAR+14] Pedro Nogueira, Rúben Aguiar, Rui Rodrigues, Eugénio Oliveira, and Lennart Nacke. Fuzzy affective player models: A physiology-based hierarchical clustering method, 2014.

[NRON13] Pedro Nogueira, Rui Rodrigues, Eugénio Oliveira, and Lennart Nacke. Guided Emotional State Regulation: Understanding and Shaping Players' Affective Experiences in Digital Games. *Aiide*, (Fairclough 2009):51–57, 2013.

[Ork06] J. Orkin. Three states and a plan: The ai of f.e.a.r. In *Proceedings of the Game Developer's Conference (GDC)*, 2006.

[OS02] Eugénio Oliveira and Luís Sarmento. Emotional valence-based mechanisms and agent personality, 2002.

[PANK02] Mannes Poel, Rieks Den Akker, Anton Nijholt, and Aard-jan Van Kesteren. Learning Emotions in Virtual Environments. *Proceedings of the Sixteenth European Meeting on Cybernetics and System Research*, pages 751–756, 2002.

[PL99] Alex Pentland and Andrew Liu. Modeling and Prediction of Human Behavior. *Neural Computation*, 11(1):229–242, 1999.

REFERENCES

[PnOPnS11] Luis Peña, Sascha Ossowski, Jose M. Peña, and Jose a. Sánchez. EEP - A lightweight emotional model: Application to RPG video game characters. *2011 IEEE Conference on Computational Intelligence and Games, CIG 2011*, pages 142–149, 2011.

[POMS05] D. Pereira, E. Oliveira, N. Moreira, and L. Sarmento. Towards an Architecture for Emotional BDI Agents. *2005 Portuguese Conference on Artificial Intelligence*, 2005.

[PTY10] Christopher Pedersen, Julian Togelius, and Georgios N. Yannakakis. Modeling player experience for content creation. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(1):54–67, 2010.

[RG95] Anand S Rao and Michael P Georgeff. *Formal models and decision procedures for multi-agent systems*. Australian Artificial Intelligence Institute Melbourne, Australia, 1995.

[Rus80] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[SDM09] Bas R. Steunebrink, Mehdi Dastani, and John-Jules Ch. Meyer. The OCC Model Revisited. *4th Workshop on Emotion and Computing*, pages 1–8, 2009.

[SM07] Bas R Steunebrink and John-jules Ch Meyer. A Logic of Emotions for Intelligent Agents. *Intelligence*, pages 142–147, 2007.

["St11] "Stuev027". OCEAN or CANOE: The Model of Personality - Section 02 & 03 F11 Psy 1001, 2011.

[TM06] D a Tavara and a Meier. Agents with personality for videogames. *Articulated Motion and Deformable Objects. 4th International Conference, AMDO 2006. Proceedings, 11-14 July 2006*, pages 484–493, 2006.

[VPB+06] H. Van Dyke Parunak, H Van Dyke Parunak, Robert Bisson, Robert Bisson, Sven Brueckner, Sven Brueckner, Robert Matthews, Robert Matthews, John Sauter, John Sauter, and Ann Arbor. A model of emotions for situated agents. *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems - AAMAS '06*, (JANUARY):993–995, 2006.

[Wal03] Scott A Wallace. *Validating complex agent behavior*. PhD thesis, Citeseer, 2003.

[Wig96] J.S. Wiggins. *The Five-factor Model of Personality: Theoretical Perspectives*. Guilford Press, 1996.

[WJ95] Michael Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(02):115–152, 1995.

[WM09] Ben G. Weber and Michael Mateas. A data mining approach to strategy prediction. *CIG2009 - 2009 IEEE Symposium on Computational Intelligence and Games*, pages 140–147, 2009.

[XLNV12] Pengfei Xing, Michael Lees, Hu Nan, and T Vaisagh Viswanthatn. Validation of agent-based simulation through human computation: an example of crowd simulation. In *Multi-Agent-Based Simulation XII*, pages 90–102. Springer, 2012.